Multimedia Systems
Lecture – 25

By

Dr. Priyambada Subudhi

Assistant Professor

IIIT Sri City

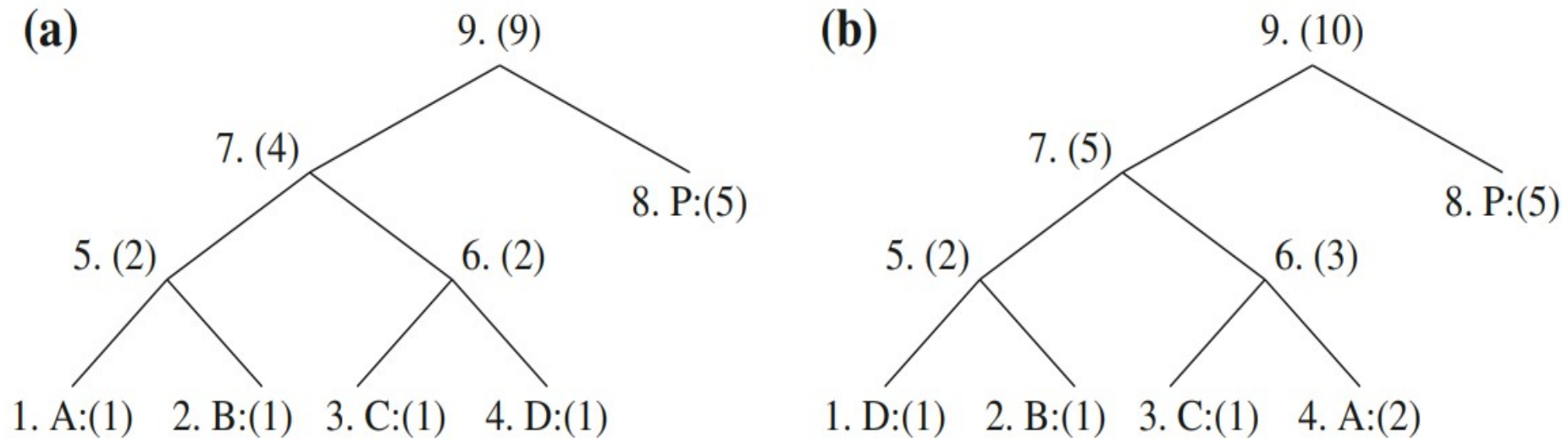
Procedure for Huffman Coding

```
ENCODER
-------

Initial_code();
while not EOF
 {
    get(c);
    encode(c);
    update_tree(c);
 }
```

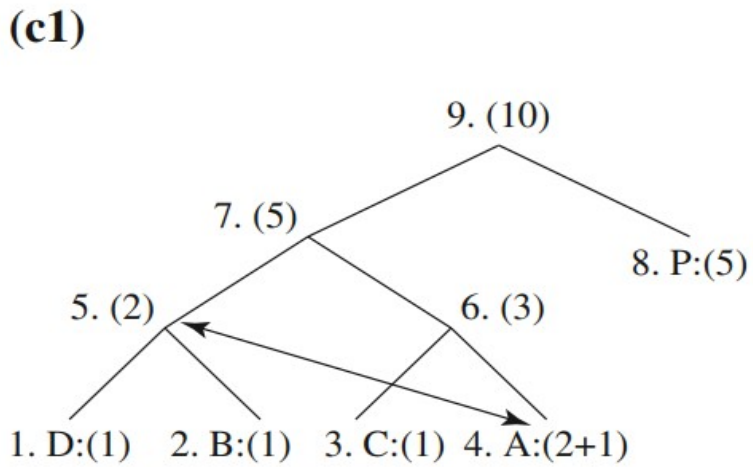
```
DECODER
-------

Initial_code();
while not EOF
   {
      decode(c);
      output(c);
      update_tree(c);
    }
```

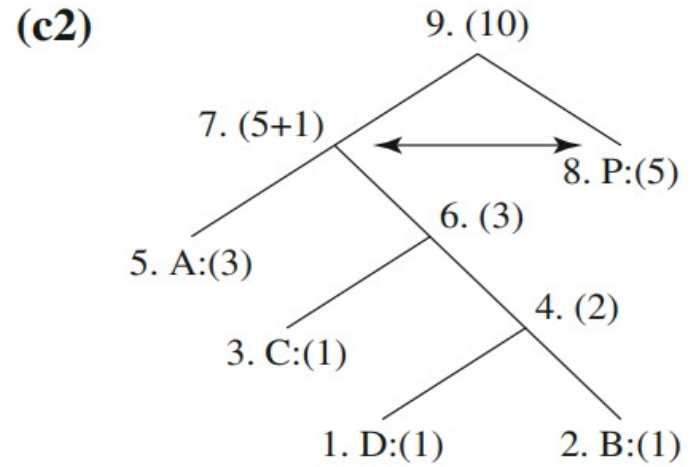
- **Initial_code** assigns symbols with some initially ***agreed-upon codes***, without any prior knowledge of the frequency counts for them. For example, some conventional codes such as ASCII may be used for coding character symbols.
- **update_tree** is a procedure for constructing an adaptive Huffman tree. It basically does two things: it increments the frequency counts for the symbols (including any new ones), and updates the configuration of the tree
  - The Huffman tree must always maintain its ***sibling property***—that is, all nodes (internal and leaf) are arranged in the order of increasing counts. Nodes are numbered in order from left to right, bottom to top. If the sibling property is about to be violated, a swap procedure is invoked to update the tree by rearranging the nodes.
  - When a swap is necessary, the farthest node with count N is swapped with the node whose count has just been increased to N + 1. Note that if the node with count N is not a leaf-node—it is the root of a subtree—the entire subtree will go with it during the swap
- The encoder and decoder must use exactly the same Initial_code and update_tree routines.

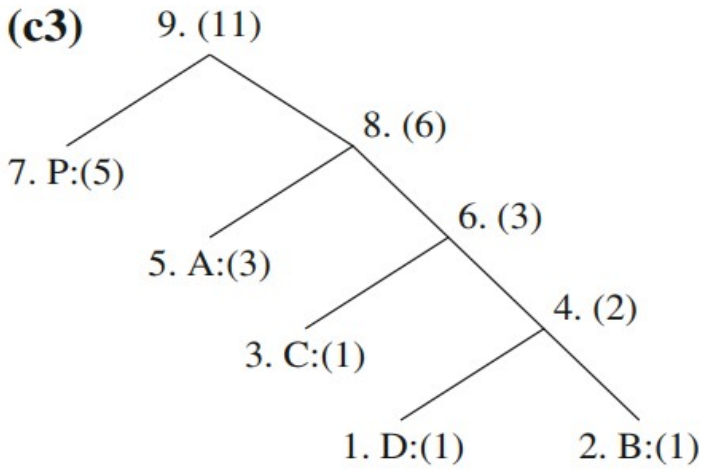
Node swapping for updating an adaptive Huffman tree: a a Huffman tree; b receiving 2nd ‘A’ triggered a swap;



c1 a swap is needed after receiving 3rd ‘A’; c2 another swap is needed; c3 the Huffman tree after receiving 3rd ‘A’



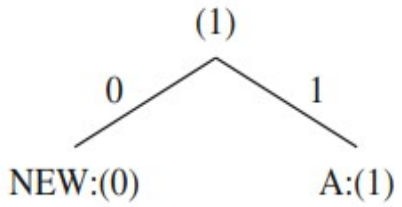




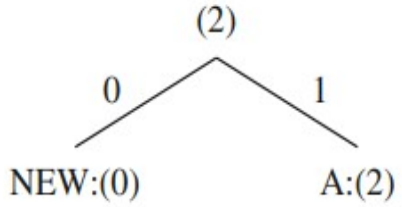
- Example: Adaptive Huffman Coding for Symbol String AADCCDD.
- Let us assume that the initial code assignment for both the encoder and decoder simply follows the ASCII order for the 26 symbols in an alphabet, A through Z.
- To improve the implementation of the algorithm, we adopt an additional rule: if any character/symbol is to be sent the first time, it must be preceded by a special symbol, NEW. The initial code for NEW is 0. The count for NEW is always kept as 0.

Symbol	Initial code
NEW	0
A	00001
B	00010
C	00011
D	00100
⋮	⋮

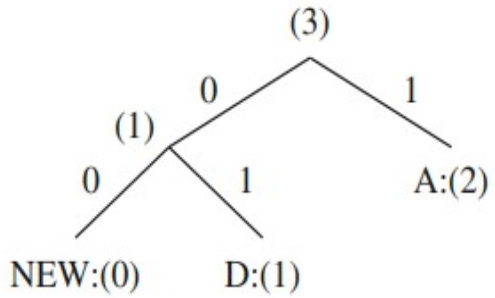
Adaptive Huffman tree for AADCCDD



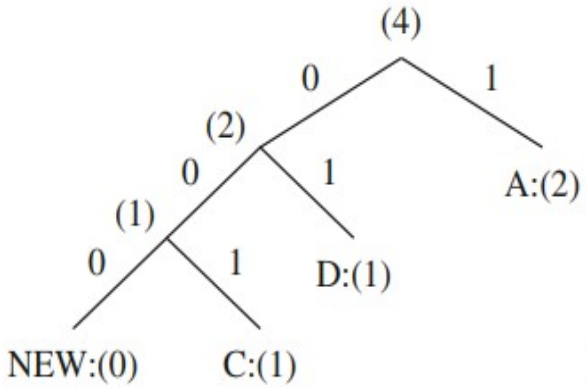
"A"



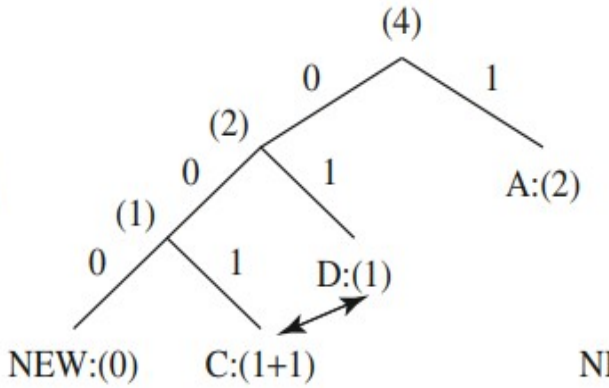
"AA"



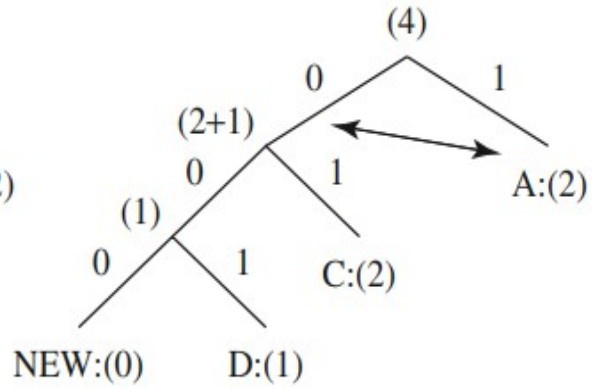
"AAD"



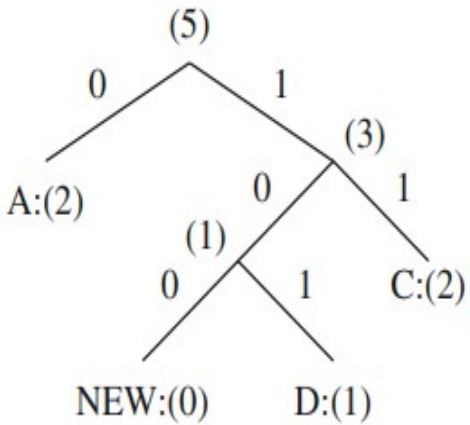
"AADC"



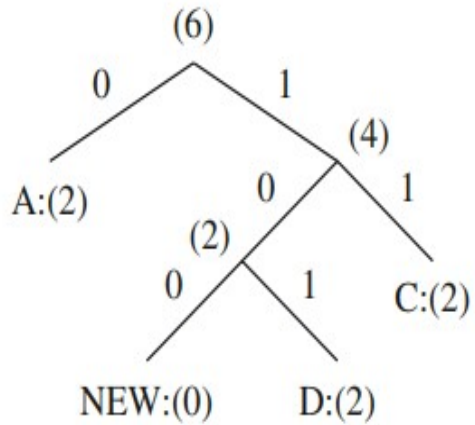
"AADCC" step 1



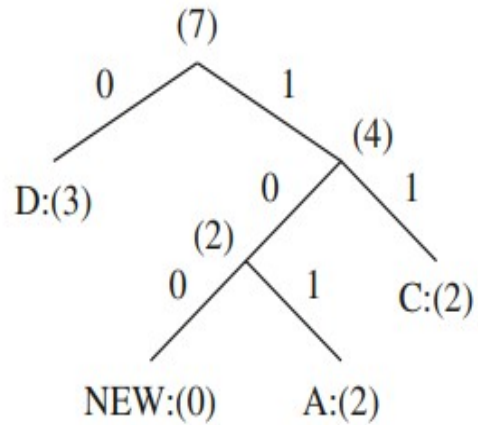
"AADCC" step 2



"AADCC" step 3



"AADCCD"



"AADCCDD"

Sequence of symbols and codes sent to the decoder

Symbol	NEW	A	A	NEW	D	NEW	C	C	D	D
Code	0	00001	1	0	00100	00	00011	001	101	101